VII International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2017
M. Papadrakakis, E. Oñate and B. Schrefler (Eds)

ON THE SIMULATION OF DROPLETIZATION

RAINALD LÖHNER[1], JOSEPH D. BAUM[2], FUMIYA TOGASHI[2], MICHAEL E. GILTRUD[2] AND ORLANDO A. SOTO[2]

[1]Center for Computational Fluid Dynamics
George Mason University, Fairfax, VA 22030, USA
e-mail: rlohner@gmu.edu, web page: https://cfd.gmu.edu/

[2]Applied Simulations, Inc., McLean, VA 22101, USA

Key words: Numerical Algorithms, Computational Fluid Dynamics, Volume of Fluid, Multiphase Flow, Combustion, Fluid-Fluid Coupling

Abstract. Compressible and near-incompressible solvers, together with particle update techniques and chemistry packages are combined in order to compute complex multiphase flows that include dropletization, vaporization and subsquent combustion.

1 INTRODUCTION

Consider the following situation: a high explosive charge is surrounded by a liquid in gas. The explosive is detonated, expanding the liquid, which then breaks up into streaks and blobs. Due to shear, these blobs then break up into droplets. The droplets also vaporize in the surrounding gas. A secondary shock impacting this mass of droplets and vapours may then lead to ignition and combustion. Some of the original liquid mass may remain as a liquid (e.g. on the ground or walls), leading to pool fires once combustion has started.
The simulation of multiphase flows combining gaseous and liquid states has received considerable attention over the last 30 years [15, 34, 36, 33, 25, 35]. Multiphase flow simulations of this type entail a number of complications as compared to single phase flows:

- The equations of state for the liquid phase tend to be complex (e.g. caviation may be present), leading to difficulties with exact or approximate Godunov solvers;
- The speed of sound in the liquid phase is much higher, requiring smaller timesteps and lengthier runs.
- The liquid may break up into droplets that evaporate and mix with the gas phase.

A number of methods have been developed in order to combine gas (compressible) and liquid (near-incompressible) solvers in a single run. The ghost method [9, 8] is a typical method. With the maturity of **single phase** flow codes, a pressing question has been how to combine gas and liquid flow models in a single run. The approach taken here was to use different flow models (and CFD codes), and to couple both via the immersed body method. In the gas region the velocities of the liquid are imposed wherever liquid is present. For the liquid region the pressures of the gas region are imposed wherever gas is present. If the liquid region can no longer be discretized via a continuum method such as the volume of fluid (VOF) or level set (LS) methods, the fluid region is converted to particles, which are incorporated into the gas region.

2 IMMERSED BODY METHOD

Consider the possibility of running concurrently solvers for gases and liquids on the same mesh. As the fluid moves through the domain, the elements and points of the grid where each one of these models is valid changes continuously. For the gas flow/ solver/ region, the **velocities** of the liquid are imposed wherever liquid exists. For the liquid flow/ solver/ region, the **pressures** of the gas are imposed wherever gas is present. In principle, both flow codes could be running concurrently on different grids. Conceptually (and implementationally) this is not different than traditional immersed body methods. Instead of having a rigidly moving body immersed in a compressible flow field, the 'body' happens to be an near-incompressible liquid.
The regions covered by the 'other fluid' for each of the fluids may be treated in a variety of ways. We have experimented with two of these, and at this point use both in production runs.
The first method uses the classic immersed body approach [11, 14, 28, 2, 7, 13, 31, 6, 30, 10, 27, 5, 20] and is sketched in Figure 1. The velocity of the gas is simply set to the velocity of the liquid wherever liquid is present. Instead of having a rigidly moving body immersed in a compressible flow field [5, 20], the 'body' happens to be the liquid. While this works well, one observes that shocks that originate in the gas phase and hit the liquid phase tend to be damped. This is to be expected, as the density and energy are still updated in the gas phase, while the velocity is artificially damped by imposing the (slow) velocity of the liquid. Another disadvantage of this approach is that unnecessary CPU is expended in the gas region covered by liquid.

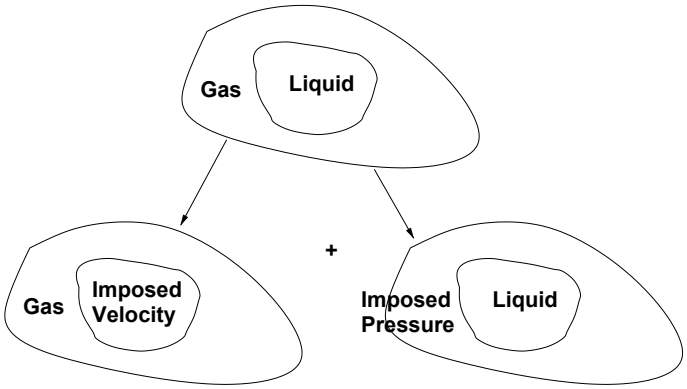


Figure 1 Immersed Body Method

The second option is to deactivate the points and edges in the gas phase covered by liquid. This leads to considerable savings in CPU, but requires the imposition of values for the end-points of edges crossing phases. At every timestep, the edges crossing phases are identified, and the values from the gas phase are extrapolated with imposed mirroring conditions to the points covered by liquids (see Figure 2). This approach is closer to the embedded surface approach [26, 12, 29, 1, 16, 22]. Instead of having a rigidly moving body surface embedded in a compressible flow field [1, 16, 21, 22], the 'body' happens to be an near-incompressible liquid.

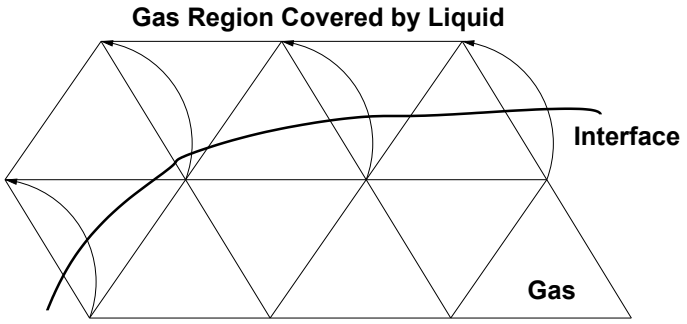


Figure 2 Extrapolation of Unknowns

3 DROPLETIZATION

The transition from liquid to blobs or large droplets uses an approach similar to large-eddy simulation: once the near-incompressible VOF flow solver can no longer discretize accurately the free surface, the badly resolved regions are transformed to blobs and transmitted to the compressible flow solvers. There, they are allowed to break up further into droplets. Several options are possible for this step. We have implemented the widely used Reitz model [32].

The question then becomes how to identify the regions that have become too 'thin' to

accurately describe the liquid via VOF or LS methods. The approach taken here is shown in Figure 3. In a first pass over the edges, the points that are fully inside, on the border of the interface or fully outside the liquid region are marked. In a second (or subsequent) pass(es), a layer is added to the points that are completely inside the liquid region. In this way, the points on the interfaces that are in the liquid but are in 'thin' regions are exposed (see Figure 3). The liquid mass at these points is subsequently transformed into droplets and passed to the code handling the gas phase, where the droplets are treated as Lagrangian particles that can exchange mass, momentum and energy with the gas.

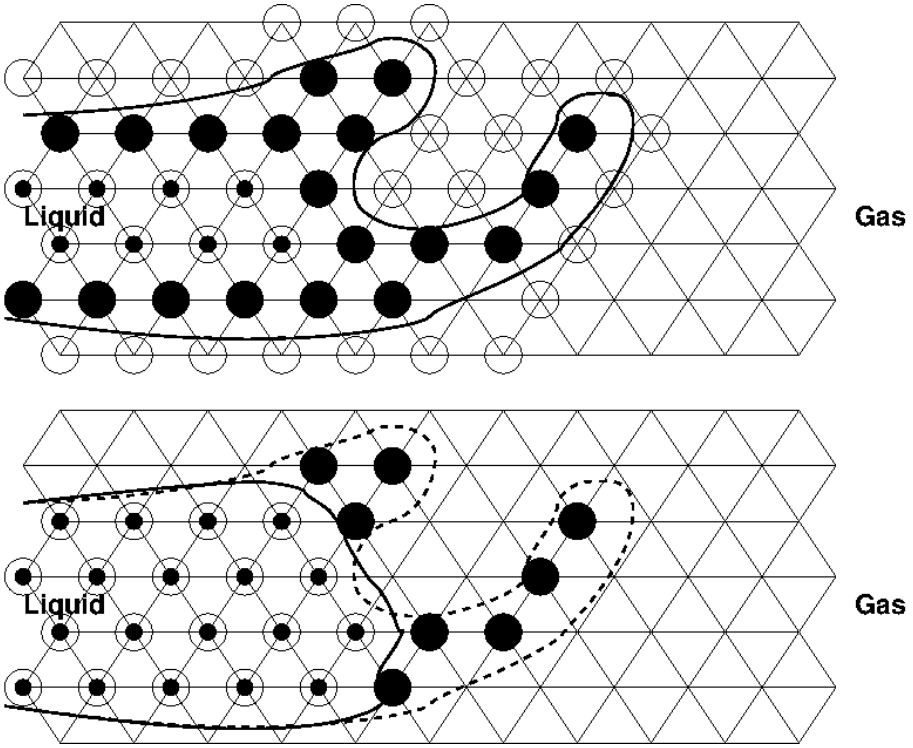
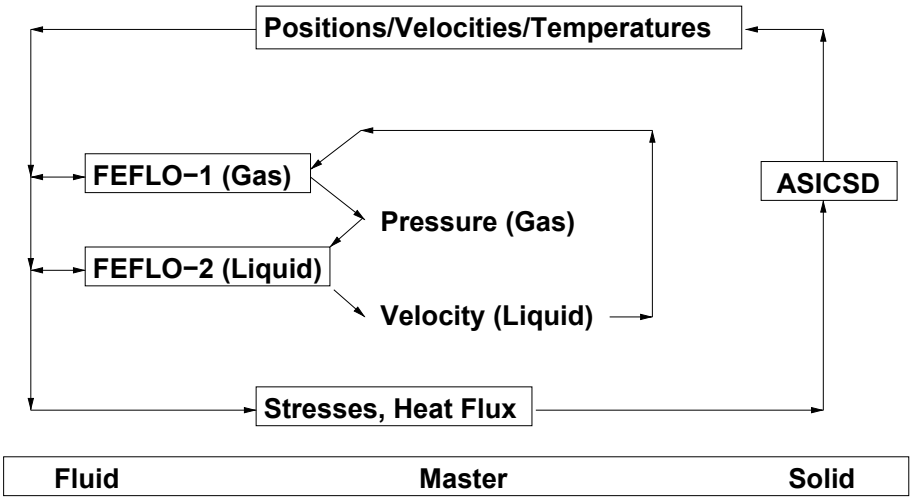


Figure 3 Dropletization of 'Thin' Regions (VOF to Particles)

4 SOFTWARE REALIZATION

The software realization of the procedure outlined above is shown in Figure 4. A master code calls the gas (compressible) and liquid (near-incompressible (+VOF)) codes, as well as the CSD code, as subroutines. The only arguments passed in and out are the (volume) unknowns to be exchanged by the flow codes, as well as the (surface) unknowns to be exchanged by the CSD code and the flow codes. The code computing the gas receives the velocities and VOF mass fraction from the code computing the liquid, updates the variables in time, and outputs the pressures. The code computing the liquid in turn receives the pressures from the code computing the gas, updates the variables in time, and outputs the velocities and VOF mass fraction. In principle, both codes could be

different. In the present case, the original flow code could be run as either compressible or near-incompressible (+VOF). Therefore, it is called twice with an 'instantiation' indicator to differentiate the input and output files required and/or generated for each domain.



S

Figure 4 Concurrent Run of Same Code Via Multiple Instantiations

A coupling of codes of this kind opens the possibility of taking different timesteps in either code. Indeed, this was pursued here, after it was realized that the solver employed to advance the liquid could run with timesteps up to two orders of magnitude larger than the solver for the gas. For the examples shown below, the timestep taken for the liquid was limited to 2 times the size of the timestep taken by the gas. This implies that, as a whole, the coupling of two codes as proposed here implies a very limited extra amount of CPU (< 3) and memory (< 2) as compared to an optimal multiphase solver.
As stated before, several solvers were combined in order to model these complex multi-physics phenomena. All of the solvers are based on unstructured grids, and use standard edge-based data structures for speed [23]. The different solvers comprise:

- For the gas part: an explicit TVD or FCT solver for chemically reacting, compressible flow [3];
- For the liquid part: a semi-explicit TVD solver for the advective-diffusive terms and a Poisson solver for the pressure terms of the near-incompressible flow [3];
- A volume of fluid approach for the free surface of the liquid [17, 18, 19, 21];
- The Chemkin [4] package for chemistry; and

- A particle update technique that allows for droplet breakup and vaporization [24].

5 RESULTS

Figure 5 shows the result of current runs. A spherical charge of TNT is detonated inside a cylinder of liquid. In order to save CPU resources without compromising the results a section of 30 degrees is considered. The mesh size is approximately 120 Mels. The evolution of the blast waves and the free surface, as well as the droplets and free surface at a late time are shown in Figure 5.

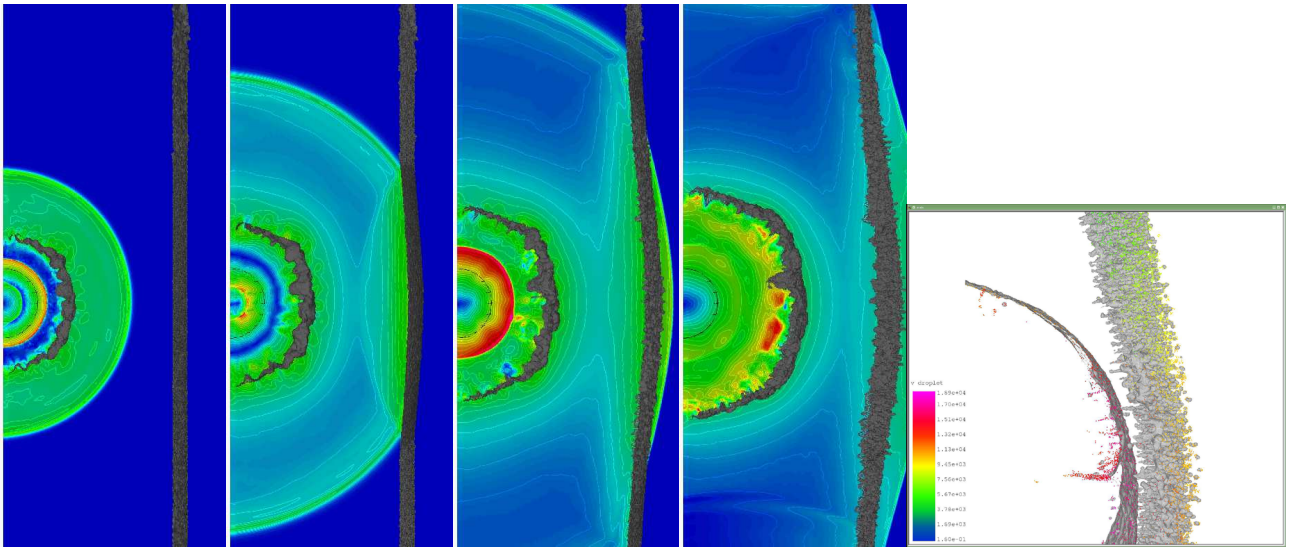


Figure 5 Evolution of Blast Wave (Velocities) And Gas/Liquid Interface

6 CONCLUSIONS AND FUTURE WORK

An approach to compute complex liquid-gas interface evolution problems has been developed. The transition from liquid to blobs uses an approach similar to large-eddy simulation: once the near-incompressible VOF flow solver can no longer discretize accurately the free surface, the badly resolved regions are transformed to blobs and transmitted to the compressible flow solvers as particles. There, they are allowed to break up further into droplets.
Future work includes extensive verification and validation, as well as porting the combination of physics modules used to massively parallel machines.

REFERENCES

[1] M.J. Aftosmis, M.J. Berger and G. Adomavicius - A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries; *AIAA*-00-0808 (2000).

[2] P. Angot, C.-H. Bruneau and P. Fabrie - A Penalization Method to Take Into Account

Obstacles in Incompressible Viscous Flows; *Numerische Mathematik* 81, 497-520 (1999).

[3] J.D. Baum, O.A. Soto, F. Togashi and R. Löhner - Numerical Modeling of Multiphase, Multimaterial Blast/Structure Interactions; *AIAA*-11-3722 (2011).

[4] R.J. Kee, F.M. Rupley and J.A. Miller - Chemkin-II: A Fortran Chemical Kinetics Package for the Analysis of Gas-Phase Chemical Kinetics; *Sandia Nat. Lab. Report SAND-89-8009, ON: DE90000917* (1989).

[5] Y. Cho, S. Boluriaan and P.J. Morris - Immersed Boundary Method for Viscous Flow Around Moving Bodies; *AIAA*-06-1089 (2006).

[6] A. Dadone and B. Grossman - An Immersed Boundary Methodology for Inviscid Flows on Cartesian Grids; *AIAA*-02-1059 (2002).

[7] E.A. Fadlun, R. Verzicco, P. Orlando and J. Moud-Yusof - Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations; *J. Comp. Phys.* 161, 33-60 (2000).

[8] Ch. Farhat, A. Rallu and S. Shankaran - A Higher-Order Generalized Ghost Fluid Method for the Poor for the Three-Dimensional Two-Phase Flow Computation of Underwater Implosions; *J. Comp. Phys.* 227, 76747700 (2008).

[9] R. Fedkiw, T. Aslam, B. Merriman and S. Osher - A Non-Oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (The Ghost Fluid Method); *J. Comp. Phys.* 152, 457-492 (1999).

[10] A. Gilmanov and F. Sotiropoulos - A Hybrid Cartesian/Immersed Boundary Method for Simulating Flows with 3-D, Geometrically Complex Moving Bodies; *J. Comp. Phys.* 207, 2, 457-492 (2005).

[11] D. Goldstein, R. Handler and L. Sirovich - Modeling a No-Slip Flow Boundary with an External Force Field; *J. Comp. Phys.* 105, 354366 (1993).

[12] S.L. Karman - SPLITFLOW: A 3-D Unstructured Cartesian/ Prismatic Grid CFD Code for Complex Geometries; *AIAA*-95-0343 (1995).

[13] J. Kim, D. Kim and H. Choi - An Immersed-Boundary Finite-Volume Method for Simulation of Flow in Complex Geometries; *J. Comp. Phys.* 171, 132-150 (2001).

[14] A.M. Landsberg and J.P. Boris - The Virtual Cell Embedding Method: A Simple Approach for Gridding Complex Geometries; *AIAA*-97-1982 (1997).

[15] R.J. LeVeque and K.M. Shyue - Two-Dimensional Front Tracking Based on High Resolution Wave Propagation Methods; *J. Comp. Phys.* 123, 354368. ((1996).

[16] Löhner, R., J.D. Baum, E. Mestreau, D. Sharov, C. Charman and D. Pelessone - Adaptive Embedded Unstructured Grid Methods; *Int. J. Num. Meth. Eng.* 60, 641-660 (2004).

[17] R. Löhner, Chi Yang, J.R. Cebral, F. Camelli, O. Soto and J. Waltz - Improving the Speed and Accuracy of Projection-Type Incompressible Flow Solvers; *Comp. Meth. Appl. Mech. Eng.* 195, 23-24, 3087-3109 (2006).

[18] R. Löhner, Chi Yang and E. Oñate - On the Simulation of Flows with Violent Free Surface Motion; *Comp. Meth. Appl. Mech. Eng.* 195, 5597-5620 (2006).

[19] R. Löhner, Chi Yang and E. Oñate - Simulation of Flows With Violent Free Surface Motion and Moving Objects Using Unstructured Grids; *Int. J. Num. Meth. Fluids* 53, 1315-1338 (2007).

[20] R. Löhner, J.D. Baum, E.L. Mestreau and D. Rice - Comparison of Body-Fitted, Embedded and Immersed 3-D Euler Predictions for Blast Loads on Columns; *AIAA*-07-1133 (2007).

[21] R. Löhner, P. Ravier, J. Roger and P. deKermel - Combination of Compressible and Incompressible Flow Codes via Immersed Methods; *AIAA*-08-0528 (2008).

[22] R. Löhner, J.R. Cebral, F.F. Camelli, S. Appanaboyina, J.D. Baum, E.L. Mestreau and O. Soto - Adaptive Embedded and Immersed Unstructured Grid Techniques; *Comp. Meth. Appl. Mech. Eng.* 197, 2173-2197 (2008).

[23] R. Löhner - *Applied CFD Techniques, 2nd Edition*; J. Wiley & Sons (2008).

[24] R. Löhner, F. Camelli, J.D. Baum, F. Togashi and O. Soto - On Mesh-Particle Techniques; *Comp. Part. Mech.* 1, 199-209 (2014).

[25] H. Luo, J.D. Baum and R. Löhner - On the Computation of Multi-Material Flows Using ALE Formulation; *J. Comp. Phys.* 194, 304-328 (2004).

[26] Melton, J.E., M.J. Berger and M.J. Aftosmis - 3-D Applications of a Cartesian Grid Euler Method; *AIAA*-93-0853-CP (1993).

[27] R. Mittal and G. Iaccarino - Immersed Boundary Methods; *Annu. Rev. Fluid Mech.* 37, 239-261 (2005).

[28] J. Mohd-Yusof - Combined Immersed-Boundary/B-Spline Methods for Simulations of Flow in Complex Geometries; *CTR Annual Research Briefs*, NASA Ames Research Center/ Stanford Univ., 317-327 (1997).

[29] R.B. Pember, J.B. Bell, P. Colella, W.Y. Crutchfield and M.L. Welcome - An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions; *J. Comp. Phys.* 120, 278 (1995).

[30] C.S. Peskin - The Immersed Boundary Method; *Acta Numerica* 11, 479-517 (2002).

[31] S. Del Pino and O. Pironneau - Fictitious Domain Methods and Freefem3d; *Proc. ECCOMAS CFD Conf.* , Swansea, Wales (2001).

[32] R.D. Reitz - Modeling Atomization Processes in High-Pressure Vaporizing Sprays; *Atomization and Spray Technology* 3, 309-337 (1987).

[33] R. Saurel and R. Abgrall - A Multiphase Godunov Method for Compressible Multifluid and Multiphase Flows; *J. Comp. Phys.* 150, 2, 425-467 (1999).

[34] K.M. Shyue - An Efficient Shock-Capturing Algorithm for Compressible Multicomponent Problems; *J. Comp. Phys.* 142, 208242 (1998).

[35] B. Wang and H. Xu - A Method Based in Riemann Problem in Tracking Multi-Material Interface on Unstructured Moving Grids; *Eng. Appl. Comp. Fluid Mech.* 1, 4, 325-336 (2007).

[36] A.B. Wardlaw and H. Mair - Spherical Solutions of an Underwater Explosion Bubble; *Shock and Vibration* 5, 89102 (1998).